



# DENTAL SHIELD

**Presenta:**

**Alan Alejandro Gutiérrez cortes**

**Eduardo Javier Bernal flores**

**Erick Arturo flores Gómez**

**Teodulfo Alegre pineda**

**Josué vela valle**

**Asesor:**

**Lic. Ignacio Sánchez Moreno**

## Contenido

<b>Introducción.....</b>	<b>2</b>
<b>Justificación .....</b>	<b>2</b>
<b>Problemática.....</b>	<b>3</b>
<b>Solución .....</b>	<b>3</b>
<b>Base de Datos .....</b>	<b>4</b>
<b>Interfaz web.....</b>	<b>5</b>
<b>Infraestructura de Red Virtualizada en AWS.....</b>	<b>8</b>
<b>Diagrama de flujo .....</b>	<b>9</b>
<b>Conclusión .....</b>	<b>10</b>
<b>Anexo A.....</b>	<b>11</b>

## Introducción

En el siguiente proyecto se abordará la creación de una aplicación web dirigida y pensada a medida para una clínica dental, la cual no contaba con un sistema para el manejo de su personal y recepción de pacientes. Es por ello se crea este sistema para cubrir la necesidad de automatizar el proceso de administración de capital humano de dicho establecimiento, así como también crear un impacto en el desahogo del proceso de recepción de pacientes, creación de citas y administración de tiempo de los médicos.

## Justificación

Este proyecto esta creado para cubrir las necesidades básicas de la clínica dental “DentalShield” como, la administración de los tiempos de los médicos para poder atender al mayor número de pacientes posibles por jornada y para los pacientes, al crear una cita específica por medio del sistema, así podrán administra el recurso de tiempo y automatizar los procesos, esto ahorrara tiempo y dinero de manera considerable.

## Problemática

la falta de un sistema para administrar las consultas y citas de los pacientes, así como los tiempos laborales de los médicos y demás personal que labora en la clínica dental “DentalShield”, también se requiere un apartado para que los clientes consulten los costos de las consultas y los procedimientos que les serán realizados o que ellos soliciten.

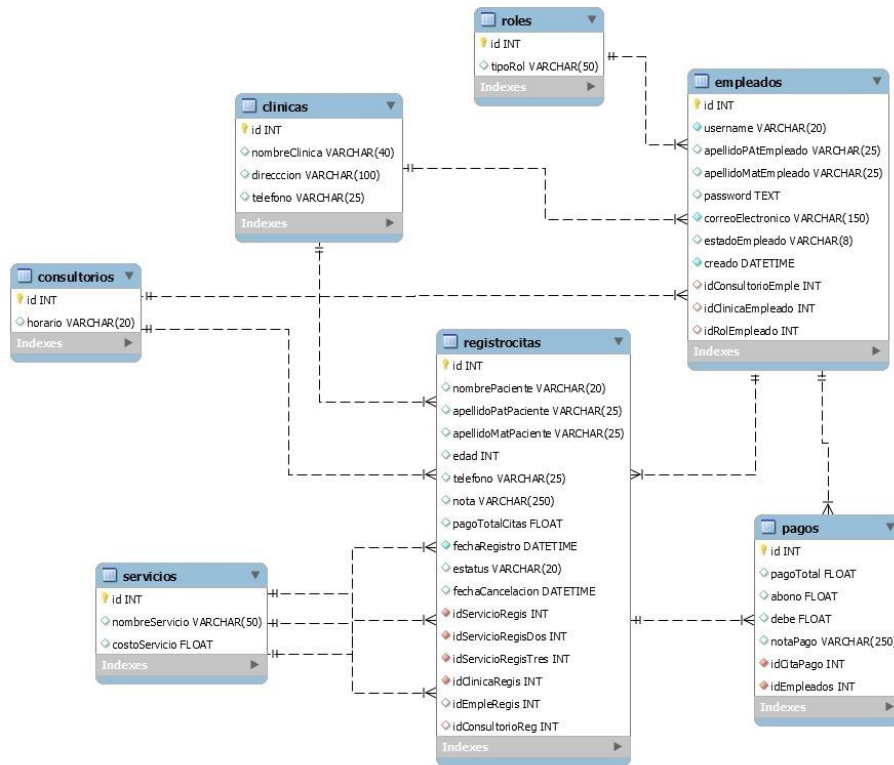
## Solución

Se crea una aplicación web alojada en un servidor de AWS para el manejo y administración de citas, médicos y costos, de la clínica dental.

Por medio de una VPC, Amazon Virtual Private Cloud (Amazon VPC) que permite lanzar recursos de AWS en una red virtual que haya definido. Esta red virtual se parece mucho a una red tradicional que operaría en su propio centro de datos, con los beneficios de utilizar la infraestructura escalable de AWS

Dicha aplicación se alojará en un servidor de AWS esto como solución viable y segura para el manejo completo de los datos que estarán siendo manipulados por los usuarios de dicha aplicación y asegurando su constante y correcto funcionamiento además de la reducción de costos al ser un servidor de arrendamiento a terceros.

## Base de Datos



En la base de datos podemos observar las siguientes tablas y sus funciones:

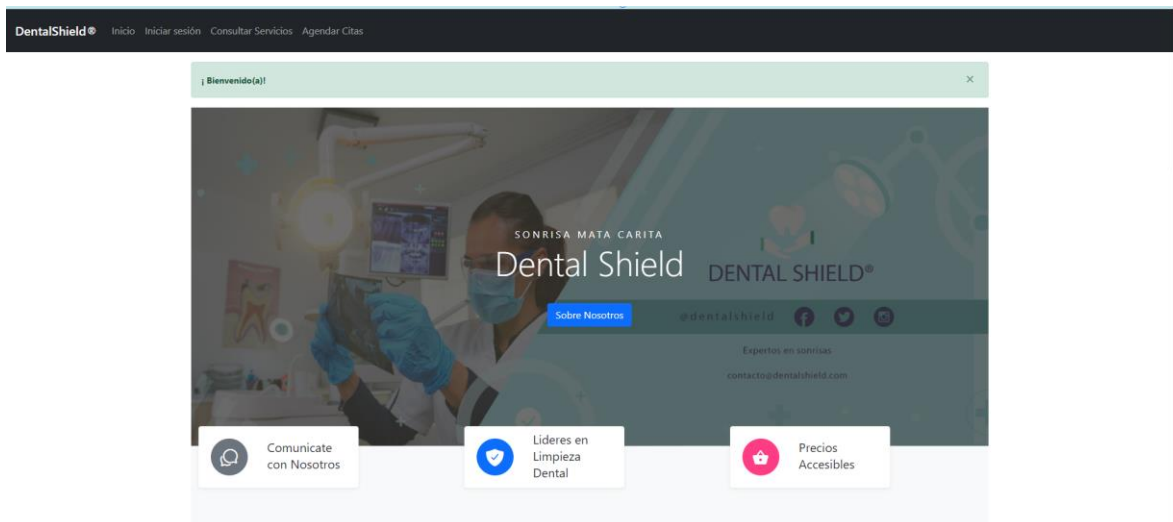
- **Empleados**: aquí podremos conocer el registro de los empleados que laboran en la clínica por medio de ID que se crea cuando se registra el nuevo empleado y sabremos su información personal.
- **Consultorio**: registramos el numero de consultorio y el horario en que esta laborando para posibles citas.
- **Servicios**: podremos seleccionar que tipo de servicio requiere el paciente (extracción, resinas, endodoncia, etc.) así como el costo de cada procedimiento.
- **Registro Citas**: los pacientes podrán acceder para generar un cita según la fecha y hora de su preferencia y sujeto a disponibilidad.
- **Cancelación**: con el id de la cita se podrá cancelar en caso de que le paciente lo requiera.
- **Roles**

**Clínicas**: llevaremos el control de cada una de las sucursales en caso de delegar a un paciente o dar información de alguna otra sucursal.

## Interfaz web

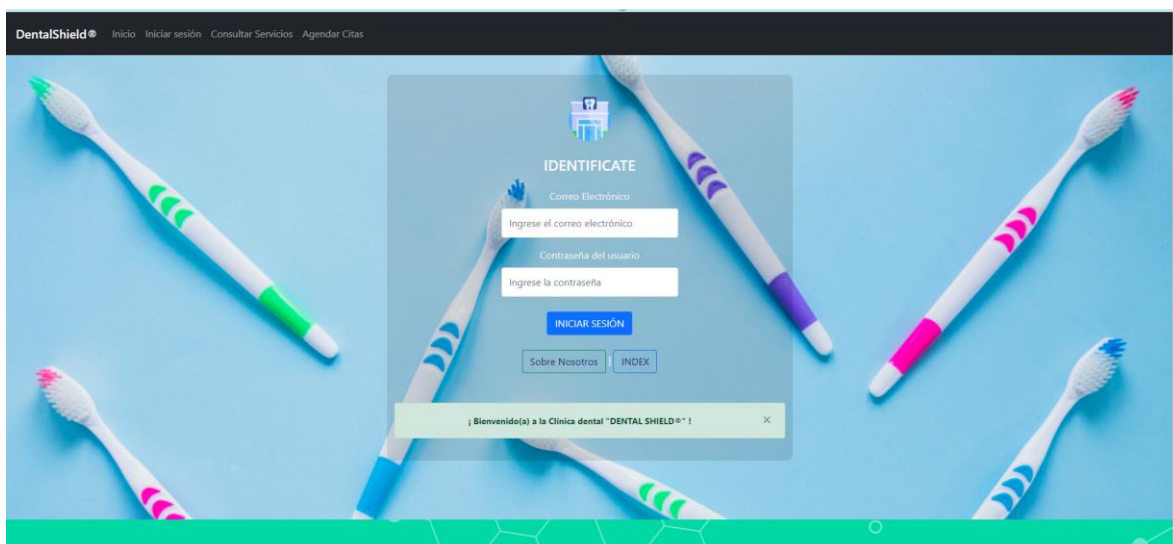
La interfaz web es la vista principal para los usuarios en general ya sean personal de la empresa o pacientes que consulten algún servicio o quieran generar una cita, a continuación, describiremos cada una de las pantallas con las que podrán interactuar los usuarios.

### 1.- página de inicio



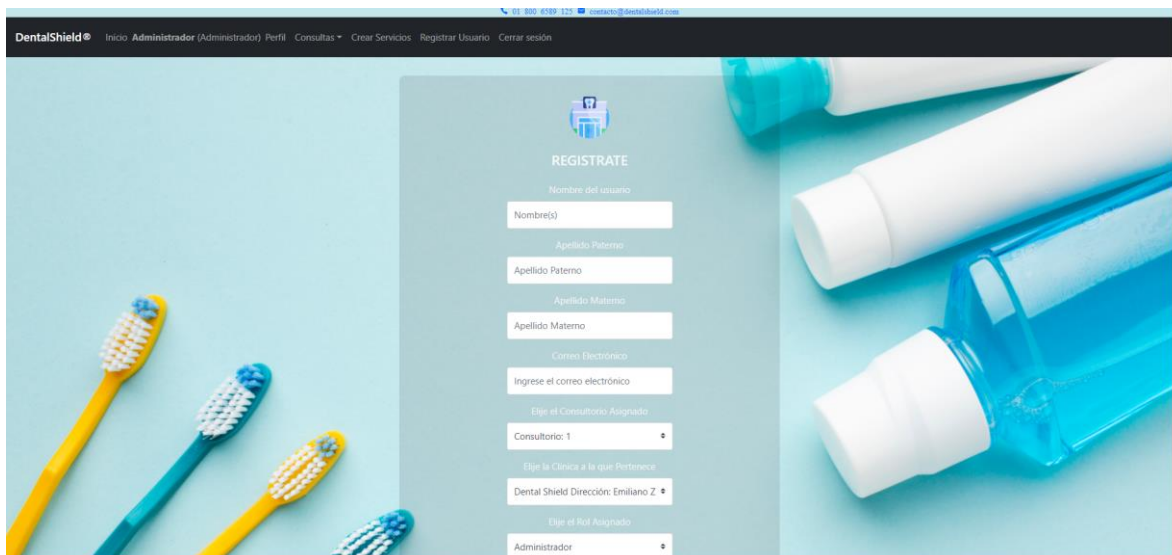
Aquí podremos observar las primeras vistas de nuestra aplicación web. Como 3 opciones principales (acerca de nosotros, precios accesibles, y comunicación.)

### 2.- Inicio de sesión.



Aquí los usuarios podrán ingresar al sistema para crear servicios, registrar nuevos usuarios, revisar consultas, pagos y citas.

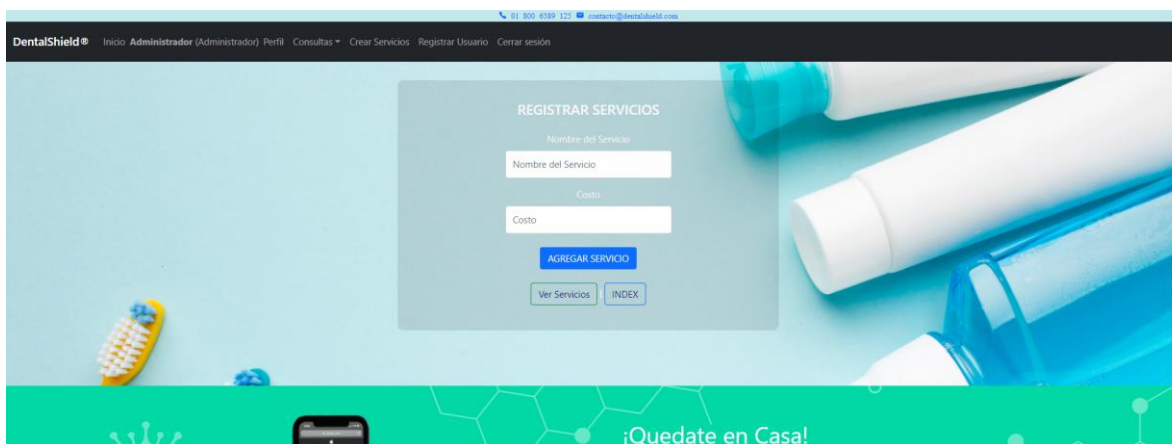
### 3.- registros



The screenshot shows the 'REGISTRATE' form in the DentalShield administrator interface. The form is set against a light blue background with dental-themed images of toothbrushes and tubes of toothpaste. The form fields include: 'Nombre del usuario' (Nombre(s)), 'Apellido Paterno', 'Apellido Materno', 'Correo Electrónico' (Ingresar el correo electrónico), 'Elegir el Consultorio Asignado' (Consultorio: 1), 'Elegir la Clínica a la que pertenece' (Dental Shield Dirección: Emiliano Z.), and 'Elegir el Rol Asignado' (Administrador). The interface also features a top navigation bar with links like 'Inicio', 'Administrador (Administrador)', 'Perfil', 'Consultas', 'Crear Servicios', 'Registrar Usuario', and 'Cerrar sesión'.

En casos de nuevos médicos que no tengan un registro en la base de datos o empleados en general en este apartado deberán llenarlo con sus datos personales para que sean parte de la base de datos y puedan ingresar al sistema.

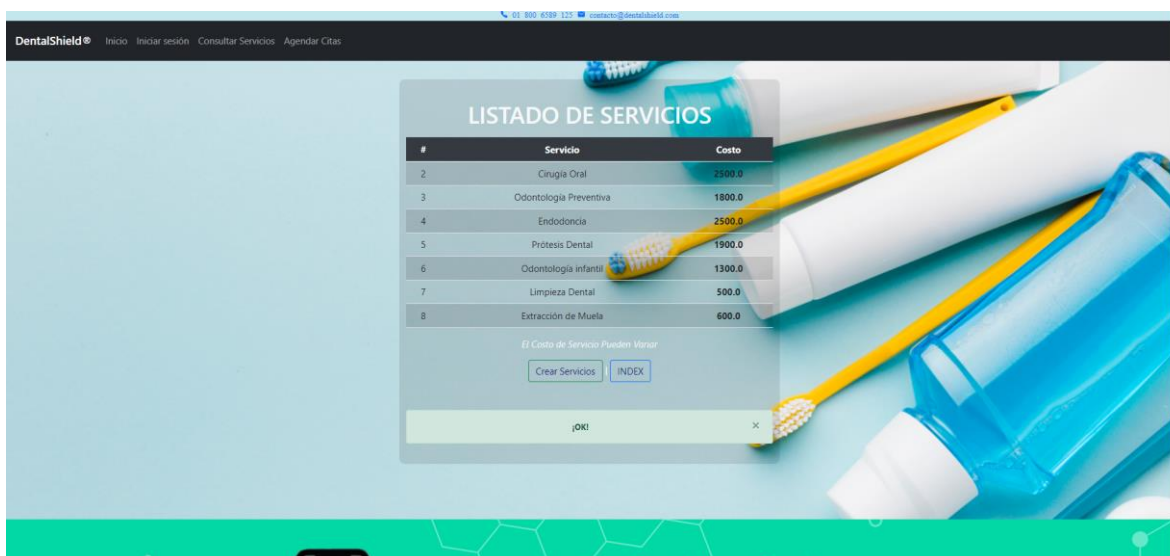
### 4.- creación de servicios



The screenshot shows the 'REGISTRAR SERVICIOS' form in the DentalShield administrator interface. The form is set against a light blue background with dental-themed images of toothbrushes and tubes of toothpaste. The form fields include: 'Nombre del Servicio' and 'Costo'. Below the fields are buttons for 'AGREGAR SERVICIO', 'Ver Servicios', and 'INDEX'. The interface also features a top navigation bar with links like 'Inicio', 'Administrador (Administrador)', 'Perfil', 'Consultas', 'Crear Servicios', 'Registrar Usuario', and 'Cerrar sesión'. At the bottom, there is a green banner with the text '¡Quedate en Casa!'.

El personal médico podrá generar nuevos servicios dependiendo de la demanda que se vaya cubriendo de los mismos, así como sus respectivos costos para que los pacientes puedan consultarlos en la página principal.

## 5.- consulta de servicios



Los pacientes podrán ver el listado de tratamientos que ofrece la clínica así como los costos de cada uno de estos.

## 6.- creación de citas

**REGISTRA UNA CITA**

Nombre del paciente

Nombre(s)

Apellido Paterno

Apellido Materno

Edad

Ej: 23

Teléfono

Ej: 7353901190

Fecha de Cita

dd/mm/aaaa

Hora de la Cita

Horario de citas 9:00am a 6:00pm

Clinicas

Los pacientes podrán agendar una nueva cita llenando los campos con sus respectivos datos esto también podrán agregar el servicio que requieren y visualizar el costo que este genera.



## Infraestructura de Red Virtualizada en AWS

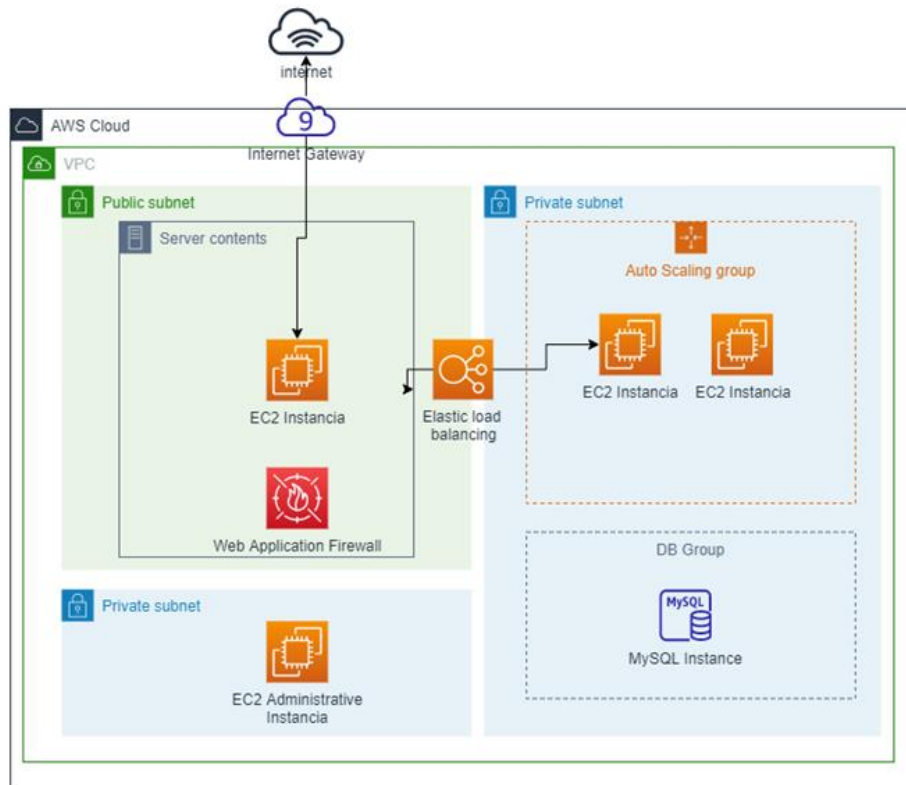


Diagrama de Estructura virtualizada

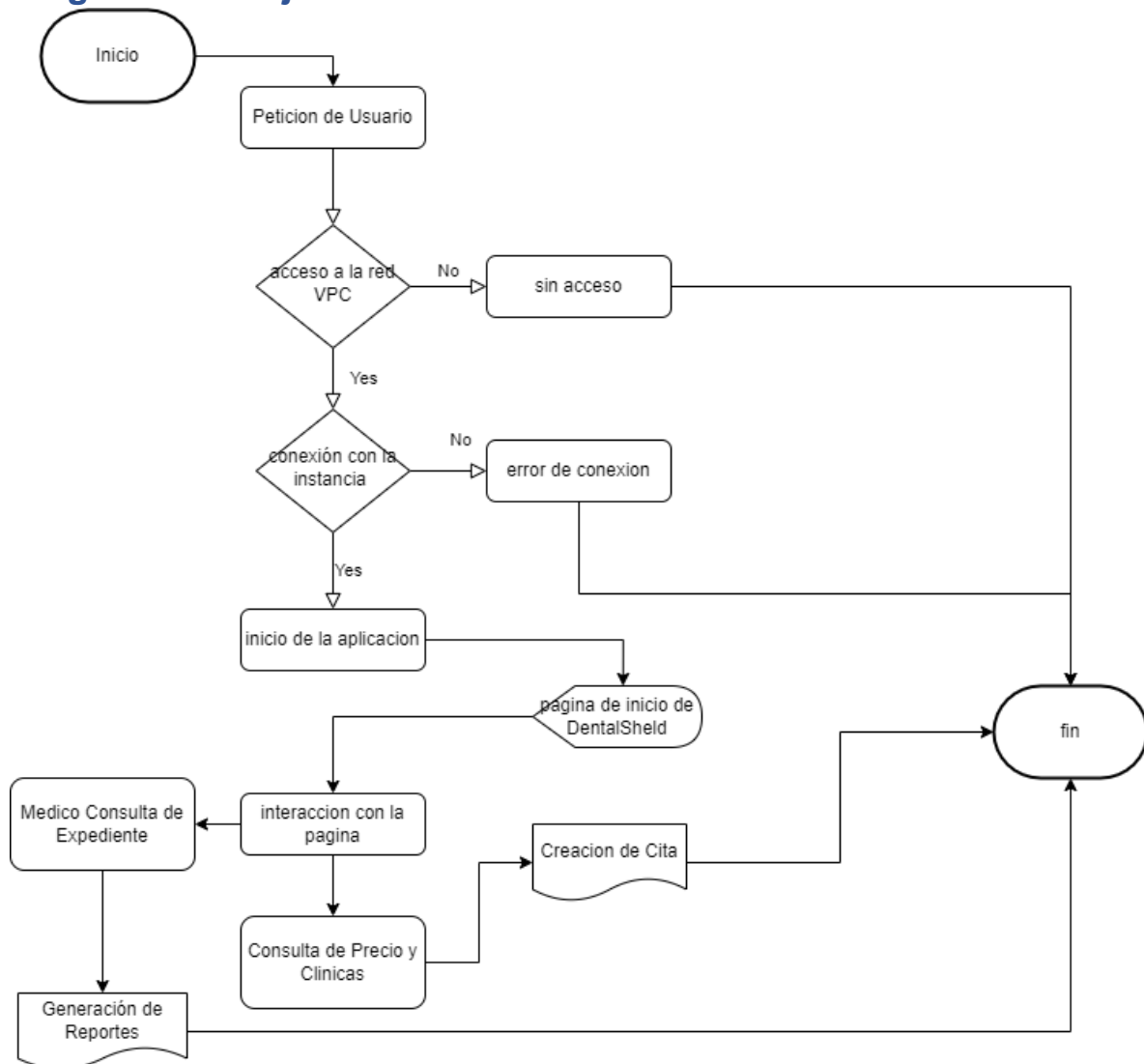
La estructura está compuesta por una red en la nube que hospeda la base de datos esta con una dirección ip privada, a la que solo se puede acceder desde la misma red de la nube.

Se configura una VPC con 4 subredes en las cuales 2 son publicas y 2 privadas. En las públicas se agregó un Gateway que tiene salida a internet y una tabla de ruteo que se divide en dos partes en una se asocian las subredes privadas y en otra las públicas.

Se configuro una instancia Ec2 en Ubuntu dentro de la misma red VPC y en esa instancia se instalaron los paquetes que necesitara la aplicación para su funcionamiento. Esta instancia está contenida en la misma red de la base de datos así que solo se tiene acceso a la base de datos desde esta instancia.

En el servicio route 53 se generaron nombres de servidores y se agregaron al apartado donde se compro el dominio para así poder usar esos nombres. Por ultimo se agrego el auto escalado que se crea a través de una snapshot en la instancia Ec2 y esto ayudara a soportar un sobrecargo de peticiones a la aplicación.

## Diagrama de flujo



Dentro del diagrama podemos observar la función básica de la aplicación web donde podemos observar, desde el inicio de la aplicación pasando por la generación de citas y reportes y la finalización de la consulta.

## Conclusión

Esta aplicación ayudara a mejorar los tiempos y administrar los pacientes así como el manejo de los expedientes medico dentales de cada uno de ellos, también llevara el control de los procedimientos que se realizan a los pacientes para que cada médico pueda tener un expediente limpio del paciente y si este no se encuentra en la clínica otro médico pueda consultar su historial médico completo, así también con este sistema podrán ver los ingresos a las clínicas dependiendo de los procedimientos que se les cobren a cada paciente y ayudara a llevar una administración contable de los ingresos totales.

El hecho de que la aplicación se hospede en un servidor cloud, garantiza la seguridad de la misma y así los datos que esta maneja y por ser un software de terceros ayuda a reducir los costos a la empresa y le da una confianza de que la aplicación estará disponible cuando se le requiera.

## Anexo A

### Script Base de Datos.

```
create database
clinicaDental; use
clinicaDental;

create table clinicas(
idClinica int,
nombreClinica varchar
(255), direcccion varchar
(255), telefono varchar
(50), constraint pk_ic
primary key (idClinica)
);
insert into clinicas values (01, "Withe Teeth", "Emiliano Zapata #1",
"7351234567"); insert into clinicas values (02, "Withe Teeth 2", "San Juan #29",
"7351234567");
#delete from clinicas where
idClinica = 05; select * from
clinicas;

create table servicios(
idServicio int,
nombreServicio varchar
(255), costoServicio float,
constraint pk_is primary
key (idServicio)
);
insert into servicios values (01, "Extraccion de
Muela", 600); insert into servicios values (02,
"Cirugía Oral", 2500); insert into servicios values (03,
"Odontología Preventiva", 1800); insert into
servicios values (04, "Endodoncia", 2500); insert into
servicios values (05, "Protesis Dental", 1900); insert
into servicios values (06, "Odontología infantil",
1300); select * from servicios;

create table roles (
idRol int, tipoRol
varchar (50),
constraint pk_ir
primary key (idRol)
```

```

);
insert into roles values (01,
"Administrador"); insert into roles values
(02, "Medico"); insert into roles values
(03, "Asistente Administrativo");
create table consultorios( claveC1 int,
horario varchar (12), constraint pk_icon
primary key (claveC1)
);
insert into consultorios values (01, 'Matutino'); insert
into consultorios values (02, 'Vespertino');

create table empleados( idEmpleado int, nombreEmpleado varchar(255), apellidoPAEmpleado
varchar(255), apellidoMatEmpleado varchar(255), cedulaProfesional varchar(255),
correoElectronico varchar(255), idConsultorioEmple int, idClinicaEmpleado int, idRolEmpleado int,
constraint pk_ie primary key (idEmpleado), constraint fk_cemp foreign key (idConsultorioEmple)
references consultorios (claveC1), #01,
02
constraint fk_ice foreign key (idClinicaEmpleado) references clinicas (idClinica), #01 constraint
fk_ire foreign key (idRolEmpleado) references roles (idRol) #01, 02, 03
);
insert into empleados values (01, "Arturo", "Flores", "Gomez", "SIN",
"arturo@witheteeth@gmail.com", 01, 01, 01);
insert into empleados values (02, "Alan", "Gutierrez", "Cortez", "01928381M",
"alan@witheteeth@gmail.com", 02, 01, 02); insert into empleados
values (03, "Javier", "Bernal", "Flores", "SIN",
"Javier@witheteeth@gmail.com", 02, 01, 01); insert into empleados values (04,
"Teodulfo", "Alegre", "Pineda", "01928482M",
"teo@witheteeth@gmail.com", 02, 01, 02); insert into empleados
values (05, "Josue", "Vela", "Valle", "SIN",
"josue@witheteeth@gmail.com", 01, 01,
01); select * from empleados;

create table registroCitas ( idCitas int, nombrePaciente varchar (255), apellidoPatPaciente
varchar (255), apellidoMatPaciente varchar (255), edad int, telefono varchar (50), estado
varchar(50), costo float, nota varchar (250), fechaRegistro datetime DEFAULT(NOW()),
idServicioRegis int, idClinicaRegis int, idConsultorioReg int, constraint pk_irc primary key
(idCitas), constraint fk_is3 foreign key (idServicioRegis) references servicios (idServicio),
constraint fk_cr foreign key (idConsultorioReg) references consultorios (claveC1), constraint
fk_clre foreign key (idClinicaRegis) references clinicas (idClinica)
);

create table cancelacion ( idCancelacion int,
telefono varchar (50), nota varchar (250),
fechaCancelacion datetime DEFAULT(NOW()),

```

```
idCitaCancelar int, constraint pk_can primary key  
(idCancelacion),  
constraint fk_ican foreign key (idCitaCancelar) references registroCitas (idCitas)  
);
```